Project report

Bitcoin prediction model

Chantoiseau Sach
Cherigui Allah Eddine
Louis Malmassary

January 2, 2025

UNIVERSITÉ
CÔTE D'AZUR

Contents

1 Introduction

The goal of this project is to develop a predictive model for Bitcoin prices using machine learning techniques. Bitcoin, being a highly volatile cryptocurrency, presents a challenging task for accurate price prediction. The primary objective is to build a model that can predict future Bitcoin prices based on historical data, thereby providing insights and aiding in decision-making for investors and traders.

2 Data Set Description

The data set used for this project consists of historical Bitcoin prices obtained from a publicly available source. The data includes daily prices of Bitcoin in USD, along with other relevant features such as the date. The data preprocessing steps involved the following:

- Reading the raw data from a CSV file.
- Converting the date column to a datetime format.
- Extracting additional features such as year and month from the date.
- Normalizing the relevant columns to ensure that the data is on a similar scale.
- Aggregating the data on a monthly basis to reduce noise and capture long-term trends.

The preprocessed data is then saved to a new CSV file for further analysis and model training.

3 Methodology

The methodology followed in this project involves several key steps:

3.1 Model Selection

Given the time series nature of the data, an LSTM (Long Short-Term Memory) neural network was chosen for the prediction task. LSTM models are well-suited for time series data as they can capture temporal dependencies and patterns.

3.2 Data Preparation

The preprocessed data is divided into time windows to create input sequences for the LSTM model. Each time window consists of a fixed number of past observations used to predict the next value in the sequence. The data is then split into training and testing sets to evaluate the model's performance.

3.3 Model Training

The LSTM model is built using the Keras library with TensorFlow as the backend. The model consists of two LSTM layers and two Dropout layers to prevent overfitting. The model is compiled with the Adam optimizer and mean squared error loss function. The training process involves fitting the model to the training data and saving the trained model and test data for evaluation.